import { StatusBar } from "expo-status-bar";

import React from "react";

import { StyleSheet, Text, View } from "react-native";

export default function App() {

  return (

    <View style={styles.container}>

      <View style={styles.viewStyleOne}>

        <Text style={styles.textStyle}> 1 </Text>

      </View>

      <View style={styles.viewStyleTwo}>

        <Text style={styles.textStyle}> 2 </Text>

      </View>

      <View style={styles.viewStyleThree}>

        <Text style={styles.textStyle}> 3 </Text>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: { backgroundColor:'#4286f4', flex:1},

  viewStyleOne: {

    width:40,

    height:40,

    justifyContent: 'center',

    alignItems:'center',

    backgroundColor:'red'

  },

  viewStyleTwo: {

    width:40,

    height:40,

    justifyContent: 'center',

    alignItems:'center',

    backgroundColor:'orange'

  },

  viewStyleThree: {

    width:40,

    height:40,

    justifyContent: 'center',

    alignItems:'center',

    backgroundColor:'green'

  },

  textStyle:{

    textAlign:'center'

  }

});

//////////////////////////////////////////////////////////////////////////////////////////////////////////////

import { StatusBar } from "expo-status-bar";

import React from "react";

import { StyleSheet, Text, View } from "react-native";

export default function App() {

  return (

    <View style={{ flex: 1 }}>

        <View style={{ flex: 0.2, backgroundColor: 'red' }} />

        <View style={{ flex: 0.2, backgroundColor: 'black' }} />

        <View style={{ flex: 0.3, backgroundColor: 'green' }} />

        <View style={{ flex: 0.3, backgroundColor: 'blue' }} />

      </View>

  );

}

const styles = StyleSheet.create({

});

//////////////////////////////////////////////////////////////////////////////////////////////////////////////

import { StatusBar } from "expo-status-bar";

import React from "react";

import { StyleSheet, Text, View } from "react-native";

export default function App() {

  return (

    <View style={ styles.container }>

        <View style={{ flex: 1, backgroundColor: 'red' }} />

        <View style={{ flex: 2, backgroundColor: 'black' }} />

        <View style={{ flex: 3, backgroundColor: 'green' }} />

        <View style={{ flex: 4, backgroundColor: 'blue' }} />

      </View>

  );

}

const styles = StyleSheet.create({

  container:{

    flex:1

  }

});

<https://micropyramid.com/blog/using-flexbox-with-react-native/>

<https://blog.reactnativecoach.com/understanding-flex-in-react-native-b34dfb4b16d1>

<https://reactnative.dev/docs/flexbox#docsNav>

<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox>

**The two axes of flexbox**

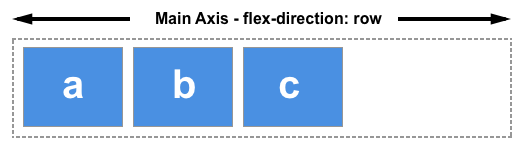
When working with flexbox you need to think in terms of two axes — the main axis and the cross axis. The main axis is defined by the [flex-direction](https://developer.mozilla.org/en-US/docs/Web/CSS/flex-direction) property, and the cross axis runs perpendicular to it. Everything we do with flexbox refers back to these axes, so it is worth understanding how they work from the outset.

The main axis

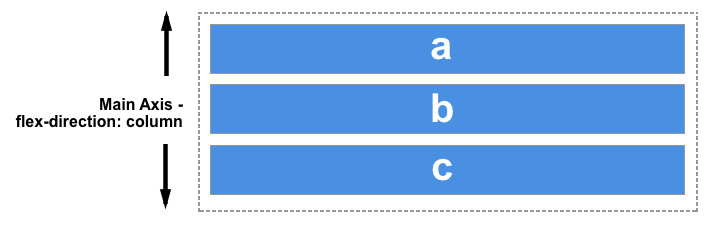
The main axis is defined by flex-direction, which has four possible values:

* row
* row-reverse
* column
* column-reverse

Should you choose row or row-reverse, your main axis will run along the row in the **inline direction**.



Choose column or column-reverse and your main axis will run from the top of the page to the bottom — in the **block direction**.

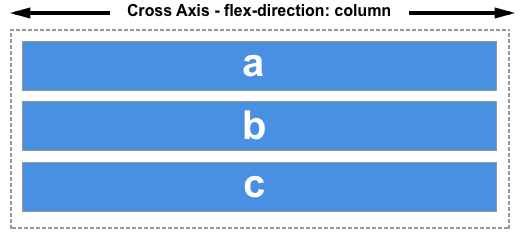


The cross axis

The cross axis runs perpendicular to the main axis, therefore if your flex-direction (main axis) is set to row or row-reverse the cross axis runs down the columns.



If your main axis is column or column-reverse then the cross axis runs along the rows.



Understanding which axis is which is important when we start to look at aligning and justifying flex items; flexbox features properties that align and justify content along one axis or the other.

///////////////////////////////////////////////////////////////////////////////////////////////////////

import { StatusBar } from "expo-status-bar";

import React from "react";

import { StyleSheet, Text, View } from "react-native";

export default function App() {

  return (

    <View style={styles.container}>

      <View style={styles.header}>

        <Text style={styles.boldText, styles.fontText}>Hello, World!</Text>

      </View>

      <View style={styles.body, styles.fontText}>

        <Text>

          Lorem ipsum <Text style={styles.boldText, styles.fontText}>dolor</Text> sit amet.

        </Text>

        <Text style={styles.fontText}>Lorem ipsum dolor sit amet.</Text>

        <Text style={styles.fontText}>Lorem ipsum dolor sit amet.</Text>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#ccc',

    alignItems: 'center',

    justifyContent: 'center',

  },

  header: {

    backgroundColor: 'pink',

    padding: 20,

  },

  body: {

    backgroundColor: 'yellow',

    padding: 20,

  },

  boldText: {

    fontWeight: 'bold',

  },

  fontText:{

    fontSize:30

  }

});

Change View in Line 11, font cannot be applied to text

///////////////////////////////////////////////////////////////////////////////////////////////////////

Using State:

import React, { useState } from 'react';

import { StyleSheet, Text, View, Button } from 'react-native';

export default function App() {

  const [name, setName] = useState('shaun');

  const [person, setPerson] = useState({ name: 'mario', age: 40 });

  const clickHandler = () => {

    setName('chun-li');

    setPerson({ name: 'luigi', age: 45 });

  };

  return (

    <View style={styles.container}>

      <Text>My name is {name}</Text>

      <Text>His name is {person.name} and his age is {person.age}</Text>

      <View style={styles.buttonContainer}>

        <Button title='update state' onPress={clickHandler} />

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

  buttonContainer: {

    marginTop: 20,

  }

});

FlatList:

import React, { useState } from 'react';

import { StyleSheet, Text, View, FlatList } from 'react-native';

export default function App() {

  const [people, setPeople] = useState([

    { name: 'shaun', id: '1' },

    { name: 'yoshi', id: '2' },

    { name: 'mario', id: '3' },

    { name: 'luigi', id: '4' },

    { name: 'peach', id: '5' },

    { name: 'toad', id: '6' },

    { name: 'bowser', id: '7' },

  ]);

  return (

    <View style={styles.container}>

      {

        people.map((item) =>{

          return(

            <View>

              <Text>{item.name}</Text>

            </View>

          )

        })

      }

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    // flex: 1,

    // paddingTop: 40,

    // paddingHorizontal: 20,

    // backgroundColor: '#fff',

  },

  // item: {

  //   flex: 1,

  //   marginHorizontal: 10,

  //   marginTop: 24,

  //   padding: 30,

  //   backgroundColor: 'pink',

  //   fontSize: 24,

  // },

});

Gives Key warning:

Change line 20 warning will go,

 <View key={item.id}>

--enables styles

Change Line 22

 <Text style={styles.item}>{item.name}</Text>

Add

{ name: 'bowser', id: '8' },

    { name: 'bowser', id: '9' },

    { name: 'bowser', id: '10' },

Add ScrollView

<ScrollView>

      {

        people.map((item) =>{

          return(

            <View key={item.id}>

              <Text style={styles.item}>{item.name}</Text>

            </View>

          )

        })

      }

      </ScrollView>

Add FlatList comment ScrollView

<FlatList

        numColumns={2}

        keyExtractor={(item) => item.id}

        data={people}

        renderItem={({ item }) => <Text style={styles.item}>{item.name}</Text>}

      />